2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Large-Scale Autonomous Vehicle Fleet Management

Timothy Mulumba[1,2]

***Abstract*— We present a hierarchical framework for city-scale autonomous ride-hailing that integrates vehicle prepositioning, request matching, charging, and facility ingress. A fine-grained mixed-integer program (MIP) coordinates prepositioning and matching on short horizons, while a coarse-grained *Deployment+Summoning* decomposition enforces charger/parking capacities at scale. On ride-hail traces, the method increases coverage and reduces wait relative to greedy and decoupled baselines, while keeping charger overuse near zero (≈1–2%) under rolling-horizon execution. We detail boundary-condition handling for 24/7 operations and specify a concrete RL training/validation protocol for a constraint-aware hybrid in which learned policies act tactically under a MIP-based safety shield.**

I. Introduction

Autonomous vehicle (AV) ride-hailing is poised to transform urban mobility, promising a future with increased safety and potentially lower operational costs. However, the transition to large-scale autonomous fleets presents a dual-use dilemma. On one hand, centrally managed AVs could optimize traffic flow, reduce energy consumption, and provide reliable transportation. On the other, a poorly managed fleet could exacerbate urban problems, increasing traffic congestion, placing undue strain on electrical grids, and creating new forms of social inequity by underserving less profitable neighborhoods. The effective management of an AV fleet is therefore not merely an operational challenge but a critical issue of public interest.

Why this matters now: Commercial driverless ride-hailing deployments and pilots have moved from small demonstrations to sustained, city-scale operations. At this scale, the fleet is a *multi-robot system* whose "actions" (dispatching, charging ingress, staging, and rebalancing) create physical congestion, queueing, and safety externalities that cities regulate and infrastructure constrains. Seemingly minor resource limits, i.e., charger stalls, depot gate throughput, curb access, and allowed operating geofences, become hard constraints that must be satisfied *in real time* while the environment changes due to demand spikes, travel-time shocks, and facility disruptions. These are control problems: high-dimensional, closed-loop decision-making under safety and resource constraints, at multiple time scales.

It requires balancing a complex set of competing goals:

- **Service Efficiency:** Providing prompt and reliable service to users by minimizing wait times and maximizing the number of fulfilled requests.
- **Environmental Sustainability:** Managing charging of a large EV fleet to operate cost-effectively and avoid overwhelming power grids.
- **Social Equity and Congestion:** Ensuring equitable access across zones while avoiding induced congestion hotspots.

A key gap in prior Autonomous Mobility on Demand (AMoD) practice is that many formulations handle *either* trip assignment/rebalancing *or* long-horizon charging/parking planning, but do not jointly enforce the hard physical constraints that dominate real deployments (limited charging capacity, ingress limits, and facility queueing), nor do they clearly specify how decisions at one time scale become constraints and targets at another. This paper is organized around making those cross-scale interfaces explicit and enforceable.

Contributions

- A unified hierarchy for AV fleet control that treats charging/parking ingress and occupancy as *hard safety/resource constraints*, enabling explainable and enforceable system-level control.
- Two interoperable workflows: (i) fine-grained *Prepositioning+Matching*, (ii) coarse-grained *Deployment+Summoning* for day-long horizons with rolling-horizon execution.
- Evaluation on city-scale traces: higher coverage, shorter waits, and near-zero charger overuse vs. greedy and decoupled baselines.
- Hybrid RL+MIP details: a concrete training/validation protocol and an on-demand MIP "safety shield" that projects learned actions onto feasible ingress/capacity sets.

Paper Outline. We next review relevant prior work and the motivations behind our decomposition approach (Section II). Section II also introduces a shared notation and an interface map linking all layers. In Sections III–VI, we detail each of our four main mathematical formulations, with an emphasis on how the model variables and constraints interact. In Section VII, we present experiments and ablations. We provide boundary-condition discussions and event-triggered replanning logic in Section VIII, then conclude in Sections IX and X with an outlook on RL extensions and real-world integration challenges.

II. Related Work and Motivation

A. Autonomous Mobility-on-Demand and MRTA.

Our problem belongs to *autonomous mobility-on-demand* (AMoD) and multi-robot task allocation (MRTA). Classic

1. New York University Abu Dhabi, UAE, 2. AI & Robotics Kampala (ARK) Lab, Kampala, Uganda. timothy.mulumba@nyu.edu

MRTA taxonomies formalize assignment structures and coupling in multi-robot systems [1], [2]. In AMoD, dynamic trip-vehicle assignment and ride-pooling at scale has been studied in high-capacity settings [3], while queueing-theoretic AMoD control, MPC, and network formulations address rebalancing and stability [4], [5]. Early work also considered queueing-network models and basic dispatch; later work incorporates uncertainty and large-scale coordination (e.g., [6]–[9]).

B. Decomposition Approaches.

Large-scale optimization often uses hierarchical decompositions [10], [11]. We similarly split AV control into a coarse Deployment layer and a finer Summoning layer, reflecting real operational cadences (hourly vs. sub-hourly). From a robotics perspective, this is a multi-time-scale control architecture: a slow "governance" planner that sets safe targets and a faster tactical layer that reacts to local changes while remaining feasible.

C. Learning-based Fleet Control

RL-based repositioning is promising but struggles with global hard constraints (e.g., facility ingress) [12]–[15]. Our stack serves as a constraint-aware baseline and strategic layer for hybrid systems. Unlike purely learning-based controllers, we make the safety envelope explicit (occupancy and ingress constraints) and provide an optimization-based projection/shield to guarantee compliance at execution time.

D. Stochasticity, Robustness, and Online Feedback

While we currently plan with deterministic forecasts, stochastic and robust approaches embed uncertainty in the model [6], [16], [17], trading tractability for resilience. Rolling horizon helps react, but lacks formal guarantees. In this paper, we make the feedback loop explicit via event-triggered re-optimization (Section VIII) and propose stress-test protocols with demand/travel-time/charging uncertainty (Section VII).

E. Fairness, Equity, and Spatial Dispersion

Ensuring equitable service across zones is increasingly central; fairness can be integrated via constraints/objectives [18], [19]. In addition to reporting spatial concentration (Gini) as a congestion proxy, we show how to incorporate dispersion directly into the fine-grained objective via an L_1 deviation regularizer (Section III), turning "fairness" from a metric into a controllable design knob.

F. Research Gap and Design Goals

We focus on a tractable, integrated framework for AV fleet management that handles interdependent layers (dispatch, charging, facility logistics) under hard physical constraints, providing a scalable foundation for governable AMoD. Concretely, we target three coupled failure modes observed in real deployments: (i) short-horizon matching that ignores battery/ingress constraints, (ii) long-horizon charging plans that do not specify which vehicles execute them, and (iii) rigid, clock-driven replanning that cannot react to shocks between ticks.

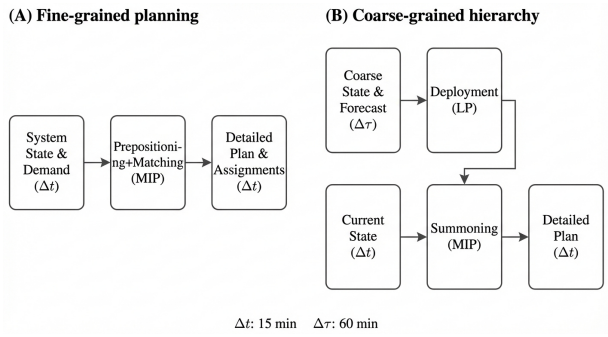

Fig. 1. Workflow overview. (A) illustrates a monolithic approach where prepositioning and matching are solved simultaneously at the fine time scale (Δt). (B) demonstrates a two-step hierarchical approach, where coarse-grained deployment targets are optimized over a longer horizon ($\Delta\tau$), which then constrain the fine-grained summoning MIP (Δt).

TABLE I

SHARED NOTATION AND INTERFACE MAP.

Quantity	Meaning (units)	Produced → Consumed
$D_{v,t}$	forecast demand in zone v at fine step t (req/step)	forecast → Preposition/Match
$Q_\tau^{\max}$	aggregate demand upper bound (req/bucket), e.g. $Q_\tau^{\max} = \sum_{t\in\tau}\sum_v D_{v,t}$	forecast → Deployment
$Y_{\tau,s,c}$	occupancy in state s with SoC bin c (#veh)	telemetry → Deployment
$N_{f,c,\tau}^{\text{prop}}$	target #vehicles (bin c) to send to facility f in bucket τ	Deployment → Summoning
$N_{f,\tau}^{\text{ing}}$	max ingress to facility f during bucket τ (#veh/bucket)	ops limits → Summoning/Shield
$x_{i,f,c,\tau}$	summoning assignment decision for vehicle i	Summoning → execution

G. Model Suite Overview

We propose two interoperable optimization frameworks (Fig. 1): a *fine-grained* Prepositioning+Matching workflow for short horizons and a *coarse-grained hierarchical* Deployment+Summoning workflow for day-scale planning.

Shared notation and interface map. We use common sets throughout: zones $\mathcal{V}$, vehicles $\mathcal{I}$, requests $\mathcal{R}$, facilities $\mathcal{F}$, charge bins $\mathcal{C}$, fine steps $t \in \mathcal{T}$ (length Δt), and coarse buckets $\tau \in \mathcal{H}$ (length $\Delta\tau$). Table I summarizes how outputs of one layer become inputs of another.

This interface map also clarifies how the hierarchy closes the loop: realized occupancies/state-of-charge (SoC) histograms update $Y_{\tau,s,c}$ and triggers re-solves when deviations exceed thresholds (Section VIII).

III. PREPOSITIONING MODEL

We start with a simpler *Prepositioning* problem, ignoring charging or facility constraints. The goal is to ensure that a sufficient number of vehicles is allocated to each location to meet predicted demand. This is sometimes done for a near-future time horizon (e.g. the next 30–60 minutes).

A. Model Definition

We are given:

- A set of locations $\mathcal{V}$ (e.g. city zones),

- A discrete time horizon $\mathcal{T} = \{1, \dots, T\}$ (time steps of length Δt),
- A set $\mathcal{K} = \{1, \dots, K\}$ representing different *demand tiers* or priority classes (e.g., premium vs. standard requests),
- A total of $|\mathcal{I}| = N_{\text{AV}}$ AVs, each introduced exactly once at some location and time.

We denote by $D_{v,t}$ the predicted demand at location $v \in \mathcal{V}$ and time $t \in \mathcal{T}$ *in requests per step* (units: req/Δt). We let β_k be the fraction of demand that belongs to tier k, and α_k be the priority weight of tier k.

Decision variables:

- $n_{v,k,t} \geq 0$: the number of AVs allocated to location v, tier k, at time t,
- $a_{i,v\to w,t} \in \{0,1\}$: equals 1 if AV i moves from location v to location w during step t,
- $s_{i,v,t} \in \{0,1\}$: equals 1 if AV i is introduced ("starts") at location v and time t,
- $d^{\text{served}}_{v,t} \geq 0$: demand served at (v,t),
- $u_{v,t} \geq 0$: unmet demand at (v,t).

Objective: We maximize tier-weighted allocation while strongly penalizing unmet demand:

$$\max \sum_{v\in\mathcal{V}} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \alpha_k \, n_{v,k,t} - \lambda \sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} u_{v,t}, \quad \lambda \gg \max_k \alpha_k. \tag{1}$$

Spatial-dispersion regularizer. To make the "congestion/fairness" proxy actionable, we can add $-\gamma \sum_{v,t} \delta_{v,t}$ to (1), where $\delta_{v,t} \geq |\sum_k n_{v,k,t} - \rho_{v,t} \sum_{v'} \sum_k n_{v',k,t}|$ and $\rho_{v,t} := D_{v,t} / \sum_{v'} D_{v',t}$. This linear L_1 penalty discourages over-concentration relative to demand [20].

Constraints:

$$\sum_k n_{v,k,t} = \sum_k n_{v,k,t-1} - \sum_{w\in\mathcal{V}} \sum_{i\in\mathcal{I}} a_{i,v\to w,t} + \sum_{w\in\mathcal{V}} \sum_{i\in\mathcal{I}} a_{i,w\to v,t} + \sum_{i\in\mathcal{I}} s_{i,v,t}, \quad \forall v,\ t > 1, \tag{2}$$

$$n_{v,k,t} \leq \beta_k \, D_{v,t}, \quad \forall k \neq K,\ v,\ t, \tag{3}$$

$$d^{\text{served}}_{v,t} + u_{v,t} = D_{v,t}, \quad \forall v,\ t, \tag{4}$$

$$d^{\text{served}}_{v,t} \leq \sum_k n_{v,k,t}, \quad \forall v,\ t, \tag{5}$$

$$\sum_{v,w\in\mathcal{V}} a_{i,v\to w,t} \leq 1, \quad \forall i,\ t, \tag{6}$$

$$\sum_{w\in\mathcal{V}} a_{i,v\to w,t} \leq \sum_{w\in\mathcal{V}} a_{i,w\to v,t-1} + s_{i,v,t}, \quad \forall i,\ v,\ t > 1, \tag{7}$$

$$\sum_{v\in\mathcal{V}} \sum_{t\in\mathcal{T}} s_{i,v,t} = 1, \quad \forall i, \tag{8}$$

$a_{i,v\to w,t}, s_{i,v,t} \in \{0,1\}$, and $n_{v,k,t}, d^{\text{served}}_{v,t}, u_{v,t} \geq 0$.

B. Explanation of Constraints

Eq. (2) (Flow Conservation). The total vehicles at location v and time t (summing across all tiers) equals the previous time step's number minus those that leave plus those that arrive plus any newly introduced vehicles. This ensures consistency of vehicle counts from one time step to the next.

Eq. (3) (Tier Fraction Cap). For each tier k except the lowest, no more than β_k fraction of the location's predicted demand can be allocated, preventing artificially inflating high-tier coverage beyond realistic demand.

Eqs. (4)–(5) (Feasible Demand Split). Instead of forcing all demand to be met, we decompose demand into served and unmet parts. The objective penalizes $u_{v,t}$ via λ, preserving feasibility even under shortages.

Eqs. (6)–(8) (Vehicle-Level Feasibility). Each vehicle makes at most one transition per time step. A vehicle may depart zone v at time t only if it was in v at $t-1$ or is newly introduced at (v,t). Each vehicle is introduced exactly once.

C. Practical Notes

This model can be solved in a rolling-horizon manner, e.g. every 15 minutes for the next 2 hours, or it can be run offline for scenario planning (e.g. for the next 24 hours). In practice, one may also add speed/distance feasibility masks so that $a_{i,v\to w,t} = 1$ only if (v,w) is reachable within Δt. The penalty λ can be tuned relative to $\{\alpha_k\}$ to prioritize serving demand while allowing limited, explicit slack when supply is insufficient.

IV. Repositioning + Matching

We now extend the above formulation to incorporate explicit *matching* of vehicles to ride requests, rather than only ensuring coverage. This is valuable if we want to track which requests are served by which vehicle, enabling direct constraints on waiting time or ride-level performance.

A. Model Definition

We add:

- A set of ride requests $\mathcal{R}$, each with pickup location/time (p_r, t_r) and an optional pickup window length $L_r \in \mathbb{Z}_{\geq 0}$ (in fine steps).
- $M^{\text{req}}_{r,w,t} \in \{0,1\}$ indicates whether request r can be served at location w and time t (e.g., $M^{\text{req}}_{r,w,t} = 1$ iff $w = p_r$ and $t \in \{t_r, \dots, t_r + L_r\}$; set $L_r = 0$ to recover exact-time service).
- $\bar{t}_{v,w}$: predicted travel time from location v to w (used to restrict feasible arcs; include $v = w$ to allow in-zone pickups).

We introduce a new decision variable:

$$m_{i,v,r,w,t} \in \{0,1\},$$

which equals 1 if vehicle i at location v is matched to request r at location w and time t. Typically, we only allow $m_{i,v,r,w,t} = 1$ if the corresponding move is selected, i.e., $a_{i,v\to w,t} = 1$.

Objective: We can maximize the number of matched requests, optionally penalizing pickup delay:

$$\max \sum_{r,i,v,w,t} \omega_r \, m_{i,v,r,w,t} - \eta \sum_{r,i,v,w,t} (t - t_r) \, m_{i,v,r,w,t}, \tag{9}$$

where $\omega_r \geq 0$ weights priorities (e.g. tiers) and $\eta \geq 0$ penalizes waiting; setting $\eta = 0$ recovers the simple "maximize matches" objective.

Additional constraints:

$$m_{i,v,r,w,t} \le a_{i,v\to w,t}, \qquad \forall i,r,v,w,t, \tag{10}$$

$$m_{i,v,r,w,t} \le M^{\text{req}}_{r,w,t}, \qquad \forall i,r,v,w,t, \tag{11}$$

$$\sum_{i,v,w,t} m_{i,v,r,w,t} \le 1, \qquad \forall r, \tag{12}$$

$$\sum_{r\in\mathcal{R}} m_{i,v,r,w,t} \le 1, \qquad \forall i,v,w,t. \tag{13}$$

Constraint (10) forces the match variable to be 0 unless the vehicle physically executes the move $v \to w$ at time t (allow $v = w$ for in-zone service). Constraint (11) ensures request r is matched only within its admissible (w, t) window. Constraint (12) ensures each request is matched at most once. Constraint (13) prevents a single vehicle from being used for multiple requests at the same exact location/time.

B. Implementation Details

The pickup window L_r induces a maximum wait of $L_r\Delta t$; choosing $L_r = 0$ returns the exact-arrival model. Travel feasibility is enforced by restricting $a_{i,v\to w,t}$ to arcs with $\bar{t}_{v,w} \le \Delta t$ (and including $v = w$). The model can be solved in real time on small instances or in short rolling horizons at city scale; when needed, a simple greedy or Hungarian-style matching can replace (9)–(13) while retaining the prepositioning plan.

V. Coarse-Grained Framework Part 1: The Deployment Model

We now present the first part of our coarse-grained hierarchical framework: the *Deployment Model*. This model takes a strategic, aggregate view, making it suitable for *longer-range planning* (e.g. a 24-hour horizon). Its purpose is to determine the optimal flow of vehicles between being in service in the city and being at various facilities (e.g. charging depots, parking lots), while considering charge levels.

A. Model Definition

Sets:

- $\mathcal{F}$: facilities (e.g. charging or parking depots),
- $\mathcal{S} = \{\text{city}\} \cup \{\text{chg}(f) : f \in \mathcal{F}\} \cup \{\text{park}(f) : f \in \mathcal{F}\}$: aggregate states (city, charge, park),
- $\mathcal{C} = \{\text{low}, \text{med}, \text{high}\}$: charge-level bins,
- $\mathcal{H} = \{1, \dots, H\}$: coarse time buckets (e.g. hours).

Parameters:

- $r^{s\to s'}_{c\to c'} \in [0, 1]$: probability that a vehicle departing state s with charge-level c and moving to s' arrives with charge-level c' (encodes charge dynamics during moves or dwell),
- $Q^{\max}_\tau$: demand upper bound (rides requested) in bucket τ,
- N^{chg}_f, N^{park}_f: capacity of charging and parking at facility f,
- N^{avail}_τ: total fleet available for control in bucket τ.
- $I_{\tau,s,c} \ge 0$: exogenous arrivals into state s at the *start* of bucket τ with bin c (e.g. vehicles returning from maintenance or coming online).
- **Consistency condition:** we require $\sum_{s,c} I_{\tau,s,c} = N^{\text{avail}}_{\tau+1} - N^{\text{avail}}_\tau$, so fleet conservation holds even when availability changes. For a constant fleet, set $N^{\text{avail}}_\tau \equiv N_{\text{AV}}$ and $I_{\tau,\cdot,\cdot} \equiv 0$.
- (optional) initial occupancies $Y^{\text{init}}_{1,s,c}$.

Decision variables:

- $Y_{\tau,s,c} \ge 0$: occupancy in state s with charge bin c in bucket τ,
- $Z_{\tau,s\to s',c} \ge 0$: departures from s to s' in bucket τ for vehicles currently in bin c (include $s' = s$ to model "stay"),
- $Q_\tau \ge 0$: rides served in bucket τ.

Objective:

$$\max \sum_{\tau=1}^{H} \alpha_\tau Q_\tau, \tag{14}$$

where α_τ can weight peak hours.

Constraints: **(i) Stock–flow consistency and fleet accounting**

$$\sum_{s'\in\mathcal{S}} Z_{\tau,s\to s',c} \le Y_{\tau,s,c}, \quad \forall \tau,\ s,\ c, \tag{15}$$

$$Y_{\tau+1,s',c'} = \sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}} Z_{\tau,s\to s',c}\, r^{s\to s'}_{c\to c'} \ +\ I_{\tau+1,s',c'}, \quad \forall \tau < H,\ s',\ c', \tag{16}$$

$$Y_{1,s,c} = Y^{\text{init}}_{1,s,c} \quad \text{(if known)}, \tag{17}$$

$$\sum_{s\in\mathcal{S}}\sum_{c\in\mathcal{C}} Y_{\tau,s,c} = N^{\text{avail}}_\tau, \quad \forall \tau. \tag{18}$$

(ii) (Service feasibility) tie served rides to available city occupancy:

$$Q_\tau \le Q^{\max}_\tau, \qquad Q_\tau \le \sum_{c\in\mathcal{C}} Y_{\tau,\text{city},c}, \quad \forall \tau. \tag{19}$$

(iii) (Facility capacities) enforce limits on occupancy:

$$\sum_{c\in\mathcal{C}} Y_{\tau,\text{chg}(f),c} \le N^{\text{chg}}_f, \qquad \forall \tau,\ f \in \mathcal{F}, \tag{20}$$

$$\sum_{c\in\mathcal{C}} Y_{\tau,\text{park}(f),c} \le N^{\text{park}}_f, \qquad \forall \tau,\ f \in \mathcal{F}. \tag{21}$$

B. Explanation of Constraints

Eqs. (15)–(18) (Stock/flow with charge transitions). At each bucket τ, total departures from (s, c) cannot exceed the current occupancy $Y_{\tau,s,c}$. The next-bucket occupancy $Y_{\tau+1,s',c'}$ equals arrivals induced by all departures (modulated by $r^{s\to s'}_{c\to c'}$) plus exogenous availability changes $I_{\tau+1,s',c'}$. Eq. (18) uses a time-varying fleet size N^{avail}_τ to remain consistent when $I \ne 0$.

Eq. (19) (Rides served). Service is bounded by demand and by the instantaneous city stock, ensuring we never claim to serve more rides than the number of in-city vehicles in bucket τ.

Eqs. (20)–(21) (Facility capacities on occupancy). Capacities are enforced on the number of vehicles present in charging/parking states, which captures physical congestion and grid interaction.

C. Usage and Rolling Horizon

Compared to the fine-grained Prepositioning+Matching model, this aggregated model scales to longer horizons (e.g. 24 hours). In practice, one solves *Deployment* hourly to plan city vs. charging vs. parking stocks and uses the *Summoning* layer to choose which specific vehicles realize those aggregate targets. Deployment outputs are converted into Summoning targets via $N^{\text{prop}}_{f,c,\tau} := Z_{\tau,\text{city}\to\text{chg}(f),c}$ (and similarly for parking), optionally adjusted by facility-specific ingress limits (Section VI).

VI. Coarse-Grained Framework Part 2: The Summoning Model

The final piece is the *Summoning* model, which picks exactly which vehicles will be routed to a facility when there is limited capacity or scheduling constraints. Suppose the *Deployment* solution indicates that 50 vehicles with low charge should go to charge during time bucket τ. But which 50 among the low-charge vehicles currently in the city, and to which charging facility? This model matches specific vehicles to the aggregate slots created by the Deployment Model.

A. Model Definition

Sets and Parameters:

- $\mathcal{I}$: candidate vehicles,
- $\mathcal{F}$: facilities, $\mathcal{C}$: charge bins, $\mathcal{H}$: coarse time buckets,
- $N^{\text{prop}}_{f,c,\tau}$: target number of vehicles (bin c) to route to facility f in bucket τ (from Deployment),
- $N^{\text{ing}}_{f,\tau}$: max number of vehicles that can *arrive* at facility f during bucket τ (ingress cap),
- $\tau_{i,f}$: travel time from vehicle i location to facility f (minutes), $\Delta\tau$: bucket length (minutes),
- $M_{i,f,\tau} \in \{0,1\}$: feasibility mask, with $M_{i,f,\tau} = 1$ iff travel allows arrival of i to f during bucket τ (e.g. $\tau_{i,f} \in [\Delta\tau(\tau-1), \Delta\tau\tau]$),
- $g_{i,f,c}$: summoning score (defined below) for sending vehicle i (bin c) to facility f.

Decision variable:

$$x_{i,f,c,\tau} \in \{0,1\},$$

which indicates if vehicle i with charge bin c is assigned to facility f in time bucket τ.

Objective:

$$\max \sum_{i\in\mathcal{I}} \sum_{f\in\mathcal{F}} \sum_{c\in\mathcal{C}} \sum_{\tau\in\mathcal{H}} g_{i,f,c}\, x_{i,f,c,\tau}. \tag{22}$$

Constraints:

$$\sum_{i\in\mathcal{I}} x_{i,f,c,\tau} \le N^{\text{prop}}_{f,c,\tau}, \qquad \forall f,c,\tau, \tag{23}$$

$$\sum_{f\in\mathcal{F}} \sum_{c\in\mathcal{C}} x_{i,f,c,\tau} \le 1, \qquad \forall i,\tau, \tag{24}$$

$$\sum_{i\in\mathcal{I}} \sum_{c\in\mathcal{C}} x_{i,f,c,\tau} \le N^{\text{ing}}_{f,\tau}, \qquad \forall f,\tau, \tag{25}$$

$$x_{i,f,c,\tau} \le M_{i,f,\tau}, \qquad \forall i,f,c,\tau. \tag{26}$$

Constraint (23) respects Deployment targets per facility/bin/bucket; (24) restricts each vehicle to at most one ingress decision per bucket; (25) enforces per-bucket ingress limits; (26) handles travel-time feasibility via a precomputed mask.

B. Summoning Score Heuristic

We capture the desirability of assigning vehicle i (with charge level l_i) to facility f through:

$$g_{i,f,c} = \frac{1}{\tau_0 + \tau_{i,f}} \cdot \Phi_{i,c} \cdot A_f, \tag{27}$$

with

$$\Phi_{i,c} = \begin{cases} \dfrac{l_0}{l_{\min,c}}\, l_i, & \text{if } l_i < l_{\min,c}, \\ l_0, & \text{if } l_{\min,c} \le l_i \le l_{\max,c}, \\ \dfrac{l_0}{100 - l_{\max,c}}\,(100 - l_i), & \text{if } l_i > l_{\max,c}, \end{cases}$$

with A_f (facility availability/priority factor), τ_0 (small offset), and l_0 (scaling parameter). The term $1/(\tau_0+\tau_{i,f})$ favors nearby facilities; $\Phi_{i,c}$ prioritizes vehicles whose SoC fits bin c; A_f promotes higher-availability or strategically weighted facilities.

C. Summoning as a Feasibility Shield (Interface to RL)

Beyond "which vehicles go where," Summoning serves as a real-time feasibility layer: any tactical controller (heuristic or RL) can propose ingress actions, and Summoning projects them onto the feasible set defined by (23)–(26). This directly addresses brittleness under facility disruptions (e.g., a charger outage) by ensuring reassignment obeys updated ingress/capacity limits without requiring a full end-to-end replanning solve.

VII. Experiments

We evaluate the proposed hierarchy on anonymized, real-world ride-hail traces. We compare against standard baselines, study boundary-condition handling, and examine scaling with fleet size and horizon length.

A. Simulation Environment and Data

We use a discrete-time, flow-based simulator built from real data. The simulator assumes (i) static, time-averaged inter-zone travel times for deterministic benchmarking, and (ii) optional stochastic travel-time and charging-delay perturbations for robustness stress tests (described below). While simplified, this choice provides a controlled, repeatable testbed to assess algorithmic logic and scalability under identical conditions.

Data and City Zones: Our dataset spans several weeks in a major metro area. We partition the city into 30 zones and generate requests at 15-minute intervals. Facilities include three charging depots (100 vehicles/hour each) and one parking facility (300 vehicles). We evaluate fleets of 500 and 2000 AVs. To address scalability beyond 10^4 vehicles, we additionally report solver scaling trends and discuss how the coarse models depend primarily on $|\mathcal{V}|$, $|\mathcal{F}|$, and $|\mathcal{C}|$ rather than $|\mathcal{I}|$ (Section VII-E).

TABLE II

PREPOSITIONING VS. NAIVE BASELINE (24HR, 500 VEHICLES, 30 ZONES). *Coverage* IS % OF TOTAL DEMAND SERVED WITHIN THE SAME ZONE/TIME BLOCK. *Wait (sec)* IS AVERAGE REQUEST WAIT TIME IN SIMULATION. *Runtime* IS SOLVER TIME IN SECONDS. RESULTS ARE MEAN ± STD OVER 10 SCENARIOS.

Method	Coverage (%)	Wait (s)	Gini	Time (s)
Naive Baseline	80.1 (±2.5)	135 (±11.2)	0.48 (±0.05)	0 -
Prepositioning	92.4 (±1.2)	97 (±5.4)	0.31 (±0.03)	84 (±8)

TABLE III

PREPOSITIONING + MATCHING (P+M) VS. GREEDY REAL-TIME. RESULTS ARE MEAN ± STD OVER 10 SCENARIOS.

Method	Coverage (%)	Wait (s)	Gini	Time (s)
Greedy	88.9 (±1.9)	112 (±8.9)	0.42 (±0.04)	0
P+M	94.0 (±0.8)	82 (±4.1)	0.28 (±0.02)	181 (±15)

Metrics and Protocol: All metrics are reported as mean ± std across 10 distinct 24-hour scenarios. In addition to standard service metrics, we report a social-impact proxy:

- Congestion (Gini Coefficient): Gini of per-zone AV density at each time, averaged over 24 hours. Values near 0 indicate evenly distributed vehicles; values near 1 indicate concentration.

We also report robustness metrics under uncertainty: (i) 95th-percentile wait, (ii) constraint violation rate (charger ingress/occupancy), and (iii) recovery time after demand spikes.

Solvers and Implementation: We use Gurobi [21] on an 8-core CPU with 64GB RAM. Unless stated otherwise, each MIP solve has a 300-second time limit. Rolling-horizon cadences follow the model design: Prepositioning/Matching (short horizon), Deployment (hourly), Summoning (sub-hourly).

B. Prepositioning-Only vs. Baseline

We first compare *Prepositioning-Only* (Section III) to a naive "no-reallocation" baseline. As shown in Table II, Prepositioning improves coverage by >10%, reduces wait time, and lowers the Gini coefficient, evidence that proactive reallocations mitigate clustering and potential congestion hotspots. When enabling the optional dispersion penalty (Section III), we observe further reductions in concentration at a modest service trade-off.

C. Prepositioning + Matching vs. Greedy Real-Time Matching

We next evaluate the fine-grained workflow. With 2000 AVs, the integrated *Prepositioning+Matching MIP* (Section IV) is solved in 2-hour rolling horizons and compared to a reactive *Greedy Matching* baseline (nearest-vehicle). Table III shows the MIP improves coverage and wait and yields a more balanced spatial distribution (lower Gini).

D. Deployment + Summoning Decomposition

We evaluate the hierarchical *Deployment* (Section V) + *Summoning* (Section VI) approach on an hourly cadence:

1) Deployment (Hourly): Solve the aggregate model to determine city/charge/park stocks and flows. From this plan, derive facility-bucket targets $N^{\text{prop}}_{f,c,\tau}$.
2) Summoning (Sub-Hourly): Assign specific vehicles to facilities, respecting ingress caps $N^{\text{ing}}_{f,\tau}$ and the feasibility mask.

With 2000 AVs, 30 zones, and up to 3 facilities, a monolithic end-to-end MILP (24 hourly steps) grew to millions of variables and did not finish within 30 minutes. The two-step hierarchy solved all 24 hours in 312s (Deployment) plus 120s (Summoning). In simulation, the two-step plan served 95% of demand at 90s average wait; the monolith, truncated at 30 minutes, served 89%.

E. Scalability with Larger Fleets

We vary fleet size from 200 to 4000. On a single 2-hour window, *Prepositioning+Matching* scales from under 5s (200 AVs) to ∼200s (4000 AVs). The *Deployment* model scales more gracefully due to aggregation, supporting day-long horizons.

Discussion Beyond 10^4 vehicles: Deployment depends on $|\mathcal{S}| \cdot |\mathcal{C}| \cdot |\mathcal{H}|$ (states × bins × buckets) and is independent of $|\mathcal{I}|$. Summoning depends linearly on the candidate set size; for $|\mathcal{I}| > 10^4$ we subsample candidates per bin/zone or use a two-stage filter (top-K per facility by $\tau_{i,f}$ and SoC urgency) to keep the MIP tractable while preserving feasibility guarantees.

F. Ablation: Partial Models vs. Full Stack

We compare partial stacks on a 24-hour, 2000-AV, 20k-request scenario. Table IV shows that while *Prepositon+Match* boosts coverage, ignoring facility constraints causes charger overuse; *Deployment-only* enforces capacities but at some service cost. The full *Deploy+Summon* stack achieves the best coverage/wait with minimal overuse and lowest congestion.

VIII. BOUNDARY CONDITIONS IN 24/7 OPERATIONS

City operations do not reset at midnight. Plans must propagate vehicle locations and state-of-charge across horizons to avoid artifacts.

A. Rolling-Horizon Continuity

At each solve (e.g., hourly or every 2 hours), initialize the state with realized occupancies and SoC from the previous horizon to maintain consistency with live operations.

B. Warm-Up and Cool-Down

Append brief warm-up/cool-down intervals around each day to produce realistic midnight states and avoid end-of-day artifacts (e.g., mass deferrals at 23:59).

TABLE IV

ABLATION ON 24HR SCENARIO WITH 2000 AVS AND FACILITY CONSTRAINTS. RESULTS ARE MEAN ± STD OVER 10 SCENARIOS.

Method	Coverage (%)	Avg Wait (s)	Charger Overuse	Congestion (Gini)	CPU Time (s)
(1) Greedy Only	86.0 (±2.1)	120 (±9.5)	16.0 (±3.5)	0.45 (±0.04)	0
(2) Preposition+Match	91.5 (±1.4)	89 (±5.1)	11.0 (±2.8)	0.33 (±0.03)	240 (±21)
(3) Deployment Only	88.2 (±1.8)	105 (±7.2)	1.0 (±0.5)	0.38 (±0.04)	60 (±5)
(4) Full Deploy+Summon	95.1 (±0.9)	78 (±4.5)	1.5 (±0.8)	0.26 (±0.02)	330 (±28)

C. Terminal State Constraints

Impose light terminal constraints (e.g., minimum city coverage or high-SoC fraction) to discourage unproductive final states.

D. Event-Triggered Replanning (Asynchronous Feedback)

To address rigid, bucket-synchronous control, we augment the nominal timers with event triggers that can initiate an early re-solve of one or more layers. Concretely, we trigger a re-solve when any of the following holds:

- **Demand shock:** the observed request rate in any zone deviates from forecast by $> \theta_D$ for m consecutive fine steps.
- **Travel-time shock:** estimated travel times exceed the planning baseline by $> \theta_T$ on a critical corridor or across a zone boundary.
- **Facility disruption:** a charging site reduces capacity (stall outage) or the estimated queue delay exceeds θ_Q.

When triggered, we warm-start the relevant solver (Deployment or Summoning) using the last feasible plan and updated telemetry, ensuring fast recovery without waiting for the next fixed tick. This yields a closed-loop controller that reacts on events while preserving hard constraint compliance through the Summoning feasibility layer.

E. Empirical Note

Without terminal constraints, solvers can exploit "free disposal" at the horizon boundary, degrading next-day performance. We recommend short horizons (1–4 hours) with real-time correction and mild terminal constraints to stabilize day-to-day behavior.

IX. MIP, RL, AND A HYBRID ARCHITECTURE FOR FLEET GOVERNANCE

A central design choice in AV fleet management is the decision engine. While this paper emphasizes mixed-integer optimization (MIP), reinforcement learning (RL) is attractive for highly stochastic, real-time decisions. We view them as complementary: MIP provides *governance and guarantees*; RL provides *adaptivity* at the tactical level.

A. RL vs. MIP: Strengths and Trade-offs

Reinforcement Learning: Strengths: Learns adaptive policies from data, capturing non-obvious behaviors (e.g., preemptive rebalancing around emerging hotspots). Challenges: Enforcing *system-wide* hard constraints is difficult (charger ingress limits, city stock caps, fairness targets). Without explicit guards, policies can violate facility or density limits, leading to grid stress or congestion spillovers.

Mixed-Integer Programming (this work): Strengths: Constraints are enforced by design (fleet size, occupancy and ingress caps, state progression), producing globally coherent plans with clear explainability. Challenges: Requires a problem model and scales with combinatorics; fine-grained, second-by-second replans can be expensive, motivating the hierarchical decomposition used here.

Scalability: RL inference can scale to $> 10^4$ vehicles when actions are aggregated at the zone/bin level, while MIP remains essential for enforcing global resource constraints. Our architecture uses MIP at coarse horizons (governance) and on-demand (safety projection), keeping solves tractable even at large scale.

Constraint-Aware Benchmark We position our models as a governance baseline: methods must match or exceed service metrics *and* adhere to charger/parking limits and spatial dispersion targets. For instance, in Table IV, greedy matching improves speed but violates charger capacity (overuse $\sim$16%), whereas our hierarchy attains top coverage/wait with minimal overuse. Any RL policy should meet the same constraint set and be evaluated under identical rollouts.

B. A Practical Hybrid: Strategy (MIP) + Tactics (RL) + Safety

We outline a concrete three-layer design that drops into our stack.

(1) Strategy Layer (hourly): Deployment as governance: Solve the *Deployment* model (Section V) hourly to produce explicit limits and targets:

- Facility-bucket targets $N^{\text{prop}}_{f,c,\tau}$ for Summoning;
- City stock budgets $\sum_c Y_{t,\texttt{city},c}$ and per-facility occupancy ceilings (Eqs. (20)–(21));
- Optional dual prices/penalties (from capacity constraints) that can shape tactical rewards.

(2) Tactics Layer (sub-hourly): RL for local actions: An RL policy $\pi_\theta(a \mid s)$ operates at 1–10 minute cadence to make short-horizon choices (e.g., fine rebalancing, pickup staging) using features such as predicted demand, local queue lengths, travel times, and the latest governance targets. Training rewards combine:

- Service terms (served requests, wait penalties);
- Movement cost (empty vehicle-miles-traveled, VMT);
- Shaping from governance: penalties for approaching ingress caps, violating occupancy ceilings, or deviating from $N^{\text{prop}}_{f,c,\tau}$.

(3) Safety/Feasibility Layer (real-time): Summoning as a shield: Before execution, pass proposed actions through a *safety filter* built on the *Summoning* MIP (Section VI):

- Action masking: disallow choices that would exceed $N^{\text{ing}}_{f,\tau}$ or violate feasibility masks;
- Projection: if the policy proposes infeasible ingress, project to the nearest feasible set via a brief Summoning solve;
- Quota tracking: enforce adherence to $N^{\text{prop}}_{f,c,\tau}$ across the bucket.

This preserves hard guarantees while retaining RL's agility.

C. Training and Validation Protocol

We specify a concrete protocol for RL training details:

- **Environment and episodes:** train in the same simulator used for evaluation, with 24-hour episodes sampled from disjoint days/weeks; reserve a held-out set for validation/testing.
- **State:** zone-level demand forecast summaries, current vehicle counts per zone and SoC bin, facility occupancy/queue estimates, remaining ingress headroom $N^{\text{ing}}_{f,\tau} - \sum_{i,c} x_{i,f,c,\tau}$, and residual quotas $N^{\text{prop}}_{f,c,\tau}$.
- **Action space (scalable):** aggregated rebalancing and staging actions at the zone/bin level (e.g., send n vehicles from zone v to w), realized via a fast assignment step; facility-ingress actions are proposed but must pass the Summoning shield.
- **Algorithm:** PPO/SAC-style actor–critic with a permutation-invariant (GNN) encoder over zones and facilities; inference is $O(|\mathcal{V}| + |\mathcal{F}|)$ per step.
- **Reward shaping and constraint handling:** start with mild penalties for constraint proximity, then anneal penalty weights upward; hard violations are prevented by the shield to avoid destabilizing learning.
- **Validation:** select checkpoints by held-out performance under identical random seeds; report both mean performance and tail metrics (P95 wait, violation rate, post-shock recovery time).

This protocol makes explicit the trade-off between RL agility and MIP guarantees: RL learns fast reactions under uncertainty, while MIP defines and enforces the feasible operating envelope.

X. Conclusion

We introduced a hierarchical framework that integrates prepositioning, matching, charging, and facility ingress for city-scale AV fleets. The *Deployment+Summoning* decomposition enforces hard capacities while scaling to day-long horizons; *Prepositioning+Matching* improves short-horizon service. On real traces, the hierarchy increases coverage, shortens waits, and keeps charger overuse near zero.

Looking ahead, a distributed or consensus-style coordination between neighboring zones/facilities (e.g., ADMM) to improve resilience under localized disruptions is promising.

References

[1] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[2] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[3] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 3, pp. 462–467, 2017.

[4] R. Zhang and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1382–1389, 2016.

[5] R. Iglesias, F. Rossi, R. Zhang, and M. Pavone, "A bcmp network approach to modeling and controlling autonomous mobility-on-demand systems," in *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016. Preprint available.

[6] A. Wallar, M. Van Der Zee, J. Alonso-Mora, and D. Rus, "Vehicle rebalancing for mobility-on-demand systems with ride-sharing," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4539–4546, IEEE, 2018.

[7] S. Wollenstein-Betech, M. Salazar, A. Houshmand, M. Pavone, I. C. Paschalidis, and C. G. Cassandras, "Routing and rebalancing intermodal autonomous mobility-on-demand systems in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12263–12275, 2021.

[8] K. Spieser, K. Treleaven, W. Zhang, E. Frazzoli, M. Pavone, and D. Bertsimas, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore," *Road Vehicle Automation*, vol. 341, pp. 229–245, 2014.

[9] M. Pavone, "Autonomous mobility-on-demand systems for future urban mobility," *Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte*, pp. 399–416, 2015.

[10] G. Desaulniers, "Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows," *Operations research*, vol. 58, no. 1, pp. 179–192, 2010.

[11] M. Rinaldi, E. Picarelli, A. D'Ariano, and F. Viti, "Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications," *Omega*, vol. 96, p. 102070, 2020.

[12] Y. Jiao, X. Tang, Z. T. Qin, S. Li, F. Zhang, H. Zhu, and J. Ye, "Real-world ride-hailing vehicle repositioning using deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 130, p. 103289, 2021.

[13] A. O. Al-Abbasi, A. Ghosh, and V. Aggarwal, "Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4714–4727, 2019.

[14] J. Xie, Y. Liu, and N. Chen, "Two-sided deep reinforcement learning for dynamic mobility-on-demand management with mixed autonomy," *Transportation Science*, vol. 57, no. 4, pp. 1019–1046, 2023.

[15] R. Ahadi, W. Ketter, J. Collins, and N. Daina, "Cooperative learning for smart charging of shared autonomous vehicle fleets," *Transportation Science*, vol. 57, no. 3, pp. 613–630, 2023.

[16] C. V. Beojone, P. Zhu, I. I. Sirmatel, and N. Geroliminis, "A hierarchical control framework for vehicle repositioning in ride-hailing systems," *Transportation Research Part C: Emerging Technologies*, vol. 168, p. 104717, 2024.

[17] J. Bi, Y. Ma, J. Zhou, W. Song, Z. Cao, Y. Wu, and J. Zhang, "Learning to handle complex constraints for vehicle routing problems," *Advances in Neural Information Processing Systems*, pp. 93479–93509, 2024.

[18] T. Sühr, A. J. Biega, M. Zehlike, K. P. Gummadi, and A. Chakraborty, "Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3082–3092, 2019.

[19] A. Kumar, Y. Vorobeychik, and W. Yeoh, "Using simple incentives to improve two-sided fairness in ridesharing systems," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 33, pp. 227–235, 2023.

[20] B. T. Thodi, T. Mulumba, and S. E. Jabari, "Noticeability versus impact in traffic signal tampering," *IEEE Access*, vol. 8, pp. 86149–86161, 2020.

[21] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2025.